Zaczynamy! Pierwsze wydanie mikroKLANU przybrało wreszcie materialną postać i zeszło na papier. Jest więc pierwsza możliwość konfrontacji oczekiwań z realizacją. W odróżnieniu od szerzącej się obecnie praktyki zwiastowania fanfarami każdej nowej iluzji, nie będziemy nikomu wmawiać — pod groźbą banicji — że oto właśnie zaspokoiliśmy wszystkie potrzeby „mikrospołeczności". Wprost przeciwnie — sądzimy, że każda publikacja będzie wykazywać jak wiele problemów pozostaje nie rozwiązanych. Chcemy jednak przełamać poczucie bezsilności i pokazać, że jednak dla chcącego... W tej dziedzinie naprawdę warto próbować — mikroinformatyka to przecież nie tylko gry telewizyjne, to także szansa na inną jakość życia. Dlatego, nie czekając beznadziejnie na „zielone światła", programy operacyjne itp. proponujemy działać „na własną rękę". My zaś będziemy pośredniczyć w wymianie doświadczeń, publikując:

— ciekawe rozwiązania konstrukcyjne
— użyteczne procedury programowe
— uwagi o eksploatacji i dodatkowych możliwościach mikrokomputerów
— opisy szczególnych właściwości popularnych układów cyfrowych
— informacje o sytuacji „rynkowej" — co, gdzie, za ile
— wiadomości o inicjatywach i wydarzeniach w sferze mikroinformatyki.

Będziemy uwzględniać różny stopień zaawansowania czytelników począwszy od tych co noszą się z zamiarem, poprzez tych co znoszą podzespoły, a skończywszy na tych, którzy wynieśli już wiele doświadczeń i obmyślają bardziej wyrafinowane rozwiązania. Forma wkładki do pisma powinna ułatwić przechowanie materiałów — do chwili, gdy właśnie okażą się potrzebne.

W mikroKLANIE będziemy zamieszczać teksty zwięzłe (do 4 stron maszynopisu). Nadsyłane rysunki (czytelne!) muszą faktycznie ułatwiać wykorzystanie pomysłu przez innych.

Redaktorem wkładki jest mgr inż. Andrzej J. Piotrowski, pracownik Przemysłowego Instytutu Elektroniki w Warszawie.

Redakcja

● Firmy polonijne są w zasadzie jedynym dostawcą sprzętu mikrokomputerowego na krajowy rynek. Żądają jednak horendalnych cen, które blokują żywiołowość rozwoju mikroinformatyki. Pobieżne porównanie wolnorynkowych cen podzespołów z ceną gotowego wyrobu zdaje się świadczyć o niesamowitym zysku firm.

Prawda wygląda jednak inaczej. Stosowane w mikrokomputerach podzespoły w większości trzeba albo sprowadzać z zagranicy, albo — co ma miejsce najczęściej — nabywać od prywatnych importerów. Ci jednak nie wydają rachunków, które firma mogłaby przedstawić Urzędowi Podatkowemu. Tak więc zakupy odbywają się z pieniędzy wpisywanych w rubrykę: zysk firmy. Kwoty te są obciążone ok. 80% podatkiem. Aby nie dokładać do interesu, firma musi żądać za układ wmontowany w sprzedawane urządenie... 5 razy więcej.

Tak więc „umiejętnie" prowadzona polityka podatkowa powoduje, że mikrokomputer kosztuje zamiast 30 — 40 tys. złotych (też jeszcze drogo) — ok. 150 tys. Ponieważ prywatnych odbiorców rzadko stać na taki wydatek, głównymi odbiorcami są... instytucje finansowane przez państwo (przeważnie uczelnie). Pieniądze krążą więc w kółko, dotacje na naukę są odpowiednio wysokie, a biurokraci mają co robić: jedni dzielą fundusze, inni obliczają podatki — wszystko oczywiście bez pomocy mikrokomputerów, które są... za drogie.

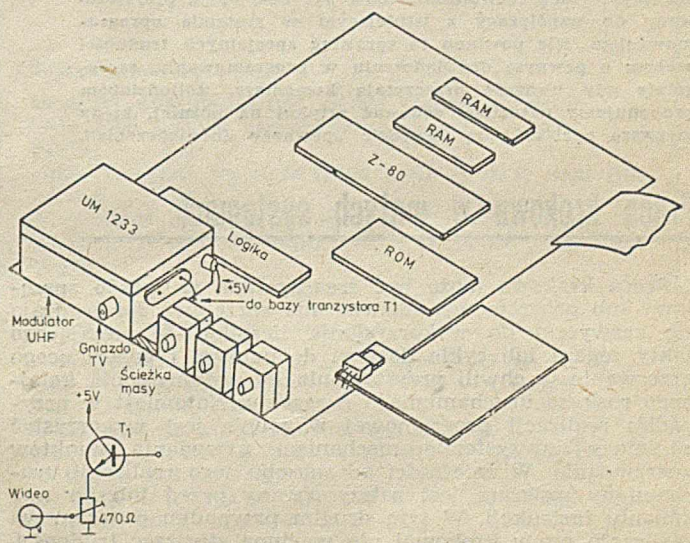
Firma polonijna AMEPROD rozpętała rewolucję mikroinformatyczną w naszym kraju, oferując ZX81 za złotówki. Cena, może nazbyt wygórowana jak na komputer osobisty, nie odstrasza jednak instytucji. Odbiorców „za własne pieniądze" wspomaga prywatny import i tu cena jest już bardziej przystępna: ZX81 z pamięcią operacyjną o pojemności 16 KB jest sporo tańszy od... radzieckiego telewizora kolorowego (a o ileż bardziej pożyteczny). Tak więc krąg użytkowników ZX81 błyskawicznie się powiększa. Publikując kolejny tekst o tym mikrokomputerze (poprzednie w nr 5 i 9/83), tym razem opisujący nietypową możliwość zastosowania, być może przyczynimy się do „odkurzenia" istniejących sieci telewizji przewodowej.

ZX81

Dzięki niewielkiemu uzupełnieniu SINCLAIR ZX81 może współpracować z systemem telewizji przewodowej. Systemy takie są dość powszechne jako wyposażenie audio-wizualnych sal wykładowych. Możliwość podłączenia mikrokomputera do istniejącej sieci telewizji przewodowej znacznie rozszerza zakres zagadnień omawianych w trakcie zajęć dydaktycznych. Przykładowo — problemy optymalizacji czy też poszukiwania różnych rozwiązań są możliwe do obliczenia w czasie jednej jednostki lekcyjnej. Graficzne przedstawienie analizowanej funkcji przyspiesza przyswojenie wykładanego tematu. Możliwe jest również wykorzystanie ZX81 do przeprowadzania testów kontrolnych, sprawdzających wiedzę studentów.

Uzupełnienie układu mikrokomputera ma na celu wyprowadzenie niezmodulowanego, zespolonego sygnału wizyjnego. Sposób zrealizowania niezbędnej przeróbki przedstawiono na rysunku. Dodatkowo zamontowano tranzystor T_1, potencjometr montażowy R_1 oraz gniazdo koncentryczne G_1.

W trakcie użytkowania ZX81 w sali wyposażonej w wielkoekranowe monitory TV należy zwrócić uwagę na zakłócenia emitowane przez układy odchylające. Nie mają one wpływu na pracę samego mikrokomputera, mogą jednak uniemożliwić poprawne wczytanie programu z taśmy, zakłócając pracę magnetofonu. Najmniej wrażliwy na te zakłócenia spośród magnetofonów krajowych jest M 101.



Szkic płytki drukowanej ZX81 z zaznaczonymi punktami podłączenia i układem dopasowującym

Ma on także inne zalety predystynujące go do współpracy z ZX81:
— licznik taśmy, ułatwiający wyszukiwanie programów na kasecie
— ręczną regulację poziomu zapisu (można uniknąć przesterowania taśmy na początku nagrywanego programu)
— układ autostopu.

Informacja na taśmie magnetofonu jest kodowana za pomocą modulowanego sygnału o częstotliwości 2,5 kHz. Stan logiczny **0** jest reprezentowany przez cztery okresy przebiegu i 1,5 ms przerwy, a stan **1** przez osiem okresów i 1,5 ms przerwy. Szybkość zapisu informacji na taśmie wynosi 300 bitów/s.

Realizacja programów w języku BASIC-ZX81 jest stosunkowo wolna. W tabeli podano czasy wykonywania operacji w trybie FAST. SLOW jest przeciętnie czterokrotnie wolniejszy, ponieważ w trakcie wykonywania programu procesor obsługuje również monitor. Długi czas realizacji potęgowania skłania do korzystania z mnożenia, o ile tylko wykładnik potęgi jest liczbą całkowitą. Czas wykonywania niektórych funkcji i instrukcji, np. VAL i DIM, zależy od wartości argumentu. Czasy podane w tabeli są wielkościami przeciętnymi, uzyskanymi z serii pomiarów.

Pamięć operacyjna ZX81 jest wykorzystana w sposób bardzo racjonalny. Nie istnieje sztywny podział na pamięć programu, danych itp. Słowo kluczowe instrukcji zajmuje tylko jeden bajt pamięci. Nieco rozrzutnie zapisywane są stałe numeryczne w tekście programu. Zajmują one tyle bajtów, ile cyfr ma dana liczba, plus 6 bajtów, w których liczba ta zapisana jest w kodzie maszynowym. Dla zmniejszenia obszaru zajmowanego przez program wskazane jest wykorzystanie zmiennych o zadanej wartości. Przynosi to oszczędność pamięci już przy trzykrotnym występowaniu tej samej stałej w programie. Dodatkowa oszczędność pamięci programu jest możliwa poprzez wykorzystanie podprogramów napisanych w kodzie maszynowym mikroprocesora Z80.

Operacja	Czas realizacji (ms)	Operacja	Czas realizacji (ms)
dodawanie	1,25	CODE	1,25
odejmowanie	1,25	CHR$	1,25
mnożenie	2	PEEK	1,25
dzielenie	3	LET	2
potęgowanie	109	(zmienna numeryczna)	
pierwiastkowanie	109		
sinus	39	GOTO	1,5
cosinus	42	IF/THEN	2,8
tangens	84	GOSUB/RETURN	2,3
arc sin	174	FOR/NEXT	4,3
arc cos	174	PRINT	140
arc tg	59	(zmienna numeryczna)	
LN	66,5		
exp	39	PRINT	1,4
INT	1,25	(zmienna tekstowa)	
SGN	0,5	POKE	2,3
ABS	0,5	PLOT	6,3
RND	11,5	UNPLOT	6,3
VAL	5	DIM	3,3
STR$	112	LET	15
LEN	1,25	(zmienna tekstowa)	

ANDRZEJ KAŁUŻA
SŁAWOMIR LITWIŃSKI
Politechnika Częstochowska

Napisanie i zasemblowanie programu to bynajmniej nie koniec pracy programisty. Trzeba jeszcze wygonić wszystkie parszywe pluskwy (ang. debug), które złośliwie namawiają Nasz Wspaniały Program do zupełnie innego funkcjonowania niż oczekujemy. Na pluskwy najlepsze jest DDT (ang. Designer Development Tool). Nie każdy jednak posiada ten środek pod ręką; natomiast znacznie łatwiej o pułapkę (nazywaną dyplomatycznie „punktem wstrzymania"). Aby sito, którym wyławiamy pluskwy było wystarczająco gęste, należałoby w zasadzie co krok zastawiać pułapkę. Stąd też zrodził się pomysł krokowej realizacji programu. Istota ułatwienia polega na automatycznym ustawianiu pułapki za każdą wykonywaną instrukcją, co oszczędza odcisków na palcach od ciągłego wprowadzania kolejnych punktów wstrzymania programu. Poniższy program, przedstawiający ideę rozwiązania, musi być oczywiście przystosowany do współpracy z istniejącym w systemie oprogramowaniem. Nie powinno to sprawiać specjalnych trudności osobom o pewnym doświadczeniu w programowaniu, szczególnie gdy uważnie przeczytają komentarz. Żółtodziobom proponujemy natomiast schować artykuł na później, kiedy pierwsze odciski będą dowodzić „pewnego doświadczenia".

Praca krokowa w małych systemach

Praca krokowa może być zrealizowana w sposób sprzętowy lub programowy. Rozwiązania sprzętowe sprowadzają się zazwyczaj do wykorzystania licznika — zliczającego takty zegara lub cykle dostępu do pamięci i generującego przerwanie w chwili przystąpienia do wykonywania kolejnego rozkazu uruchamianego programu. Natomiast w przypadku realizacji programowej wygodnie jest wykorzystać istniejący w systemie mechanizm ustawiania punktów wstrzymania. W zależności od sposobu jego realizacji uruchamiany program jest zatrzymywany przed lub po wykonaniu instrukcji. W tym drugim przypadku programowa realizacja pracy krokowej nie powinna stwarzać trudności.

Poniżej przedstawiono sposób realizacji pracy krokowej z wykorzystaniem punktów wstrzymania, zatrzymujących uruchamiany program przed wykonaniem wskazanej instrukcji i zabezpieczających zachowanie niezmienionego stanu środowiska programu. Zaprezentowany algorytm może zostać zrealizowany dla różnych mikroprocesorów. W przypadku 8080 uzyskano kod wynikowy o niewielkiej długości (ok. stu bajtów), co umożliwia wykorzystanie go w niedużych systemach mikroprocesorowych o ograniczonej pamięci stałej.

Prezentowany program można łatwo zmodyfikować dla uzyskania dodatkowych możliwości, takich jak: wykonanie określonej liczby instrukcji w jednym kroku, wykonanie programu z zachowanym śladem określonej liczby ostatnio wykonywanych instrukcji itp.

Wykorzystywane w programie punkty wstrzymania powinny być ustawione na adres instrukcji, do której może być przekazane sterowanie po wykonaniu kroku. Ponieważ instrukcje warunkowe umożliwiają rozejście na dwie drogi, wykorzystywane są dwa punkty wstrzymania. Jeżeli do dyspozycji jest tylko jeden punkt wstrzymania, to należy badać rodzaj warunku i sprawdzać jego spełnienie w celu wyboru właściwej drogi. Punkty wstrzymania muszą być chwilowe, tzn. powinny być usuwane po zatrzymaniu programu na jednym z nich. Zestaw instrukcji 8080 zawiera dziewięć różnych możliwości przekazywania sterowania po wykonaniu dyrektywy (tab).

Typ instrukcji	Pierwszy możliwy adres następnej instrukcji	Drugi możliwy adres następnej instrukcji
Jednobajtowe (bez PCHL, RET, RST)	PC+1	—
Dwubajtowe	PC+2	—
Trzybajtowe (bez JMP i CALL)	PC+3	—
JMP i CALL bezwarunkowe	(PC+1)	—
JMP i CALL warunkowe	(PC+1)	PC+3
PCHL	HL	—
RET bezwarunkowe	(SP)	—
RET warunkowe	(SP)	PC+1
RST	(PC) AND 38H	—

Niektóre możliwości przekazywania sterowania, podane w tabeli, mogą być obsługiwane w identyczny sposób, bowiem ustawione dodatkowo i nie wykorzystane punkty wstrzymania po wykonaniu kroku zostaną usunięte. Pozwala to na połączenie kilku pozycji wymienionych w tabeli w jedną grupę instrukcji. Klasyfikację kodu instrukcji do jednej z tych grup można zrealizować w różny sposób. Metoda zastosowana w przedstawianym przykładzie polega na porównywaniu kodu rozkazu (po wyzerowaniu pól argumentów) z kodami przechowywanymi w tablicy. Długość tablicy można znacznie zmniejszyć łącząc kody różniące się jednym bitem we wspólną pozycję tablicy. Np. instrukcję IN o kodzie 0DBH i instrukcję OUT o kodzie 0D3H można badać łącznie uznając bit różniący te kody za bit argumentu. Z tablicy można także wyłączyć najbardziej liczną z grup, gdyż po zbadaniu, że kod nie należy do żadnej z pozostałych grup, można go bez dalszego badania zaklasyfikować do grupy nie objętej tablicą.

Każda pozycja tablicy zawiera maskę, przez którą instrukcja jest mnożona logicznie, oraz wzorzec, z którym jest porównywana. W rejestrze b ustawiany jest numer pozycji tablicy, umożliwiający wybór odpowiedniego sposobu ustawienia punktów wstrzymania. Adres punktu wstrzymania jest dostępny w parze rejestrów HL.

```
1: ;        REALIZACJA PRACY KROKOWEJ DLA MIKROPROCESORA 8080
2: ;
3: ;        ZMIENNE ZEWNETRZNE: PCBANK — ZACHOWANY PC PROGRAMU
4: ;                            SPBANK — ZACHOWANY SP PROGRAMU
5: ;                            HLBANK — ZACHOWANY HL PROGRAMU
6: ;        PUNKT WEJSCIA DO STARTU PROGRAMU — GO
7: ;
8: SSTEP:   LHLD PCBANK     ; POBRANIE KODU ROZKAZU DO C
9:          MOV  C, M
10:         MVI  B, 11      ; PORÓWNANIE Z TABELA KODÓW
11:         LXI  H, TABCOD
12: DECOD:  MOV  A, C       ; MNOŻENIE KODU PRZEZ MASKĘ
13:         ANA  M
14:         INX  H          ; PORÓWNANIE Z KODEM TABELI
15:         CMP  M
16:         JZ   RSTN       ; SKOK JEŻELI ROZPOZNANY KOD
17:         DCR  B
18:         INX  H
19:         JNZ  DECOD      ; SKOK JEŻELI NIE KONIEC TABELI
20: ;
21: ;       KOD INSTRUKCJI JEDNOBAJTOWEJ
22: ;
23:         LHLD PCBANK
24:         INX  H
25: ; ***   USTAW PUNKT WSTRZYMANIA (HL = PC+1)
26:         LHLD SPBANK
27:         MOV  E, M
28:         INX  H
29:         MOV  D, M
30:         XCHG
31:         JMP  SETBP      ; HL = (SP)
32: ;
33: ;       INSTRUKCJA RST
34: ;
35: RSTN:   DCR  B
36:         JNZ  PCHLO      ; SKOK JEŻELI NIE RST N ;
37:         MOV  A, C
38:         ANI  38H
39:         MVI  H, Ø
40:         MOV  L, A
41:         JMP  SETBP      ; HL = N*8
42: ;
43: ;       INSTRUKCJA PCHL
44: ;
45: PCHLO:  DCR  B
46:         JNZ  INSB2      ; SKOK JEŻELI NIE PCHL
47:         LHLD HLBANK
48:         JMP  SETBP      ; HL = HL
49: ;
50: ;       INSTRUKCJE DWUBAJTOWE
51: ;
52: INSB2:  LHLD PCBANK
53:         MOV  A, B
54:         SUI  4
55:         JNC  INSB3      ; SKOK JEŻELI NIE DWUBAJTOWA INSTR.
56:         INX  H
57:         INX  H
58:         JMP  SETBP      ; HL = PC+2
59: ;
60: ;       INSTRUKCJE TRZYBAJTOWE
61: ;
62: INSB3:  INX  H
63:         MOV  E, M
64:         INX  H
65:         MOV  D, M
66:         INX  H
67:         JZ   SETBP      ; HL = PC+3
68:         PUSH H
69:         XCHG
70: ; ***   USTAW PUNKT WSTRZYMANIA HL = (PC+1)
71:         POP  H
72: SETBP:  ; *** USTAW PUNKT WSTRZYMANIA
73:         JMP  GO         ; START URUCHAMIANEGO PROGRAMU
74: TABCOD: DB   ØFFH, ØC3H ; JMP, B = 11
75:         DB   ØFFH, ØCDH ; CALL, B = 1Ø
76:         DB   ØC7H, ØC4H ; C, B = 9
77:         DB   ØC7H, ØC2H ; J, B = 8
78:         DB   ØCFH, ØØ1H ; LXI, B = 7
79:         DB   ØE7H, Ø22H ; STA, LDA, LHLD, SHLD, B = 6
80:         DB   ØC7H, ØØ6H ; MVI, B = 5
81:         DB   ØC7H, ØC6H ; ADI, ACI, SUI, SBI,
82:                         ; ORI, CPI, ANI, XRI, B = 4
83:         DB   ØF7H, ØD3H ; IN, OUT, B = 3
84:         DB   ØFFH, ØE9H ; PCHL, B = 2
85:         DB   ØC7H, ØC7H ; RST, B = 1
86:         END
```

IRENEUSZ MYZIK
Przemysłowy Instytut Elektroniki
Warszawa

Ceny • Ceny • Ceny • Ceny • Ceny

listopad—grudzień 1983

Mikrokomputery:

RFN: ZX SPECTRUM (16K) — 163 $; (48K) — 201 $
USA: ZX81 (1K) (do samodzielnego montażu) — 30 $
Polska:
Firma UNITEX: ZX SPECTRUM (16K) — 320 $; ZX81 (1K) — 146 $
Z ogłoszenia (ceny wywoławcze): ZX81 (16K) — 90 tys. zł; ZX SPECTRUM (16K) — 180 tys. zł; DRAGON — 500 „zielonych"

Podzespoły na Perskim Jarmarku:

zestaw8080, 8224, 8228 — 3,2 tys. zł; 8085 — 4,5 tys. zł
Z80A — 4 tys. zł; Z80B — 5 tys. zł; 6800 — 4 tys. zł
2708 — 1 tys. zł; 2716 — 2,5 tys. zł
8x4116 — 15 tys. zł; 8x4164 — 60 tys. zł
8251 — 1,2 tys. zł; 8255 — 2 tys. zł; 8279 — 4 tys. zł; 8257 — 4,5 tys. zł; 8755 — 4,5 tys. zł;
Z80 DMA — 3,5 tys. zł; Z80 PIO — 3,5 tys. zł; Z80 SIO/2 — 3,5 tys. zł.

Żaden mikrokomputer nie może obejść się bez pamięci RAM (ang. Random Access Memory). Jest to jednak jeden z najdroższych składników mikrokomputera. Dlatego też, oprócz wymyślania coraz to doskonalszych mikrokomputerów, „jajogłowi" poświęcają wiele swego cennego czasu właśnie pamięciom.

Pamięci dynamiczne są rozwiązaniem pozwalającym stosunkowo małym kosztem upchnąć sporo informacji w jednej „kostce" (są już pamięci o pojemności 256 KB). Róża — jednak nie bez kolców. Zasadniczą wadą DRAM-ów (ang. Dynamic Random Access Memory) jest konieczność regularnego odświeżania zawartości pamięci. Niektóre firmy oferują specjalizowane układy „załatwiające" problem współpracy procesora z pamięciami dynamicznymi (np. INTEL 8202). Jednak cena 8202, przewyższająca kilkakrotnie cenę CPU, skutecznie eliminuje stosowanie tego układu w kraju. Układy innych firm też nie są tanie.

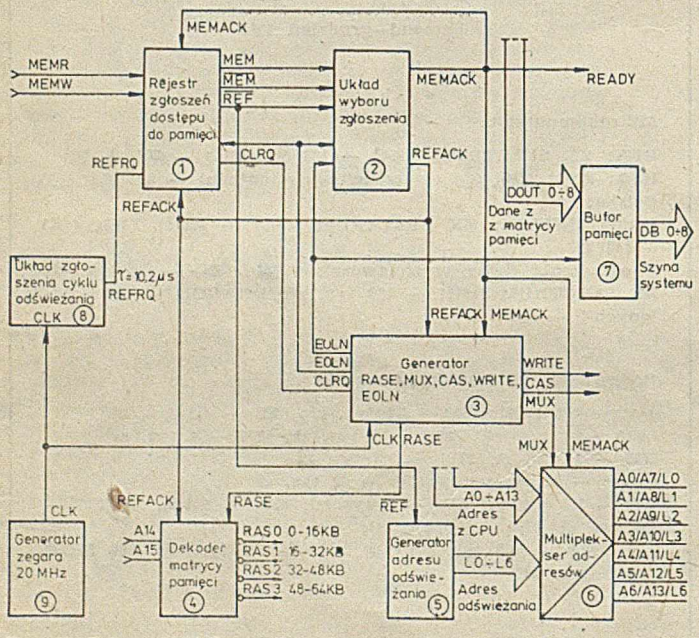
Zamieszczony poniżej opis konstrukcji zakłada, że czytelnik posiada już pewien zasób informacji o DRAM-ach. Co będzie, jeśli okaże się, że zbyt wysoko ustawiliśmy poprzeczkę — trudno, będziemy wspólnie narzekać na brak literatury fachowej! Ponieważ CEMI zapowiedziało wprowadzenie do produkcji w 1983 roku pamięci dynamicznej 16 KB (którego to dzisiaj mamy...?), chyba niebawem każdy szanujący się mikrokomputer „dostanie" swoje 64 KB.

A zatem jak zrealizować:

Sterownik pamięci dynamicznych

Schemat blokowy przykładowego sterownika pamięci dynamicznej zbudowanego z układów TTL przedsta;wiono na rysunku 1. W skład sterownika wchodzą następujące bloki funkcjonalne:

1) rejestr zgłoszeń dostępu do pamięci
2) układ wyboru zgłoszenia
3) układ generatora strobów: RASE, MUX, CAS, WRITE, EOLN (ang. End of line)
4) dekoder bloków matrycy pamięci dynamicznej
5) generator adresu odświeżania
6) multiplekser adresów
7) bufor pamięci
8) układ zgłoszenia cyklu odświeżającego
9) generator zegara 20 MHz.



Rys. 1. Schemat blokowy interfejsu pamięci dynamicznej

Generator zegara został zbudowany przy użyciu układu 74S04 i dostarcza impulsy zegarowe o częstotliwości 20 MHz, które umożliwiają generowanie sygnałów RASE, MUX, CAS, WRITE, EOLN oraz sygnału zgłoszenia cyklu odświeżania REFRQ (ang. Refresh Request). Schemat ideowy generatora zegara przedstawiony jest na rysunku 3 (blok 9). Na tym samym rysunku przedstawiono układ zgłoszenia cyklu odświeżającego (blok 8). Proponowane rozwiązanie zapewnia odświeżanie jednego wiersza pamięci co 10,2 µs, czyli każdy bit pamięci odświeżany jest co 1,3 ms.

Układ zgłoszenia cyklu odświeżania składa się z dwóch liczników 74LS193. Impulsy zegarowe CLK o częstotliwości 20 MHz dzielone są wstępnie przez 16 (pierwszy licznik — U4). Wyjście TCU (ang. Terminal Count Up) z pierwszego licznika doprowadzone jest do wejścia zegarowego CLK drugiego licznika (U4). Na wyjściu TCU licznika U3 pojawia się impuls co 256 taktów zegara. Impuls ten wykorzystywany jest jako sygnał REFRQ.

Sterownik pamięci dynamicznej może znajdować się w jednym z trzech stanów.

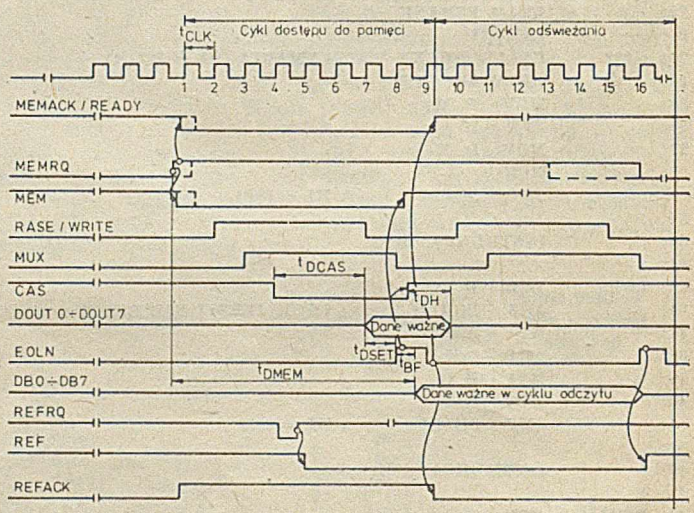
— cyklu zapisu/odczytu pamięci przez CPU
— cyklu odświeżania zawartości pamięci
— cyklu oczekiwania na zgłoszenie cyklu 1 lub 2.

Rejestr zgłoszenia dostępu (blok 1), przechowuje zgłoszenia MEMRQ (ang. Memory Request) oraz REFRQ (ang. Refresh Request). Układ wyboru zgłoszenia (blok 2), decyduje o kolejności obsługi żądań zapamiętanych w rejestrze zgłoszenia dostępu.

Żądania MEM (ang. Memory) i REF (ang. Refresh) obsługiwane są w kolejności zgłoszenia, zgodnie z zasadą FIFS (ng. First In First Served). W przypadku zgłoszenia któregokolwiek z żądań podczas obsługi poprzedniego, jego obsługa jest zawieszona do momentu zakończenia aktualnie wykonywanego cyklu.

Sygnał $\overline{\text{READY}}=0$ generowany jest tylko wtedy, gdy sterownik przejdzie do wykonywania cyklu zapisu/odczytu z pamięci tzn. $\overline{\text{MEMACK}}=0$ (ang. Memory Acknowledge). Każdy cykl zakończony jest anulowaniem odpowiedniego żądania w rejestrze zgłoszenia dostępu. W cyklu oczekiwania układ wyboru zgłoszenia generuje sygnały $\overline{\text{MEMACK}}=1$ $\overline{\text{REFACK}}=0$.

Sterownik pamięci dynamicznej rozpoczyna generowanie sygnałów strobujących do matrycy pamięci w przypadku, gdy sygnał CLRQ=1 (ang. Cycle Request). Kolejność po-



Rys. 2. Przebiegi czasowe w układzie interfejsu pamięci DRAM w systemie µP
(t_{CLK} — 50ns; t_{DSET} — min 20ns; t_{DCAS} — max 180ns; t_{DMEM} — min 350ns, max 403ns; t_{BF} — max 28ns; t_{DH} — min 0, max 40ns)

jawiania się sygnałów strobujących oraz zależności czasowe między nimi obrazuje rysunek 2. Schemat ideowy generatora strobów (blok 3) przedstawia zaś rysunek 3.

Sygnał RASE umożliwia generowanie strobów RAS0÷÷RAS3 w sposób następujący:

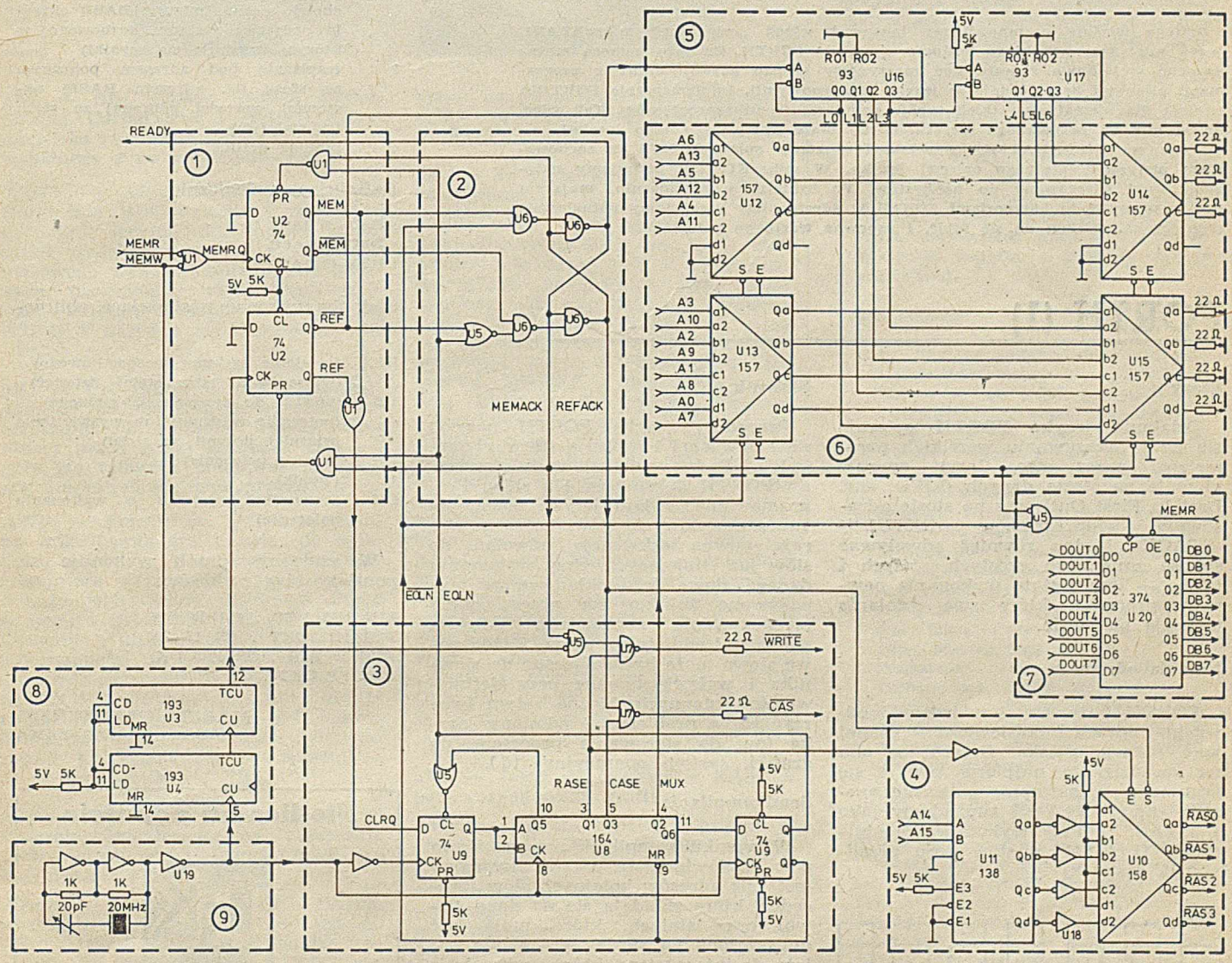
RASE	REF/ ACK	A15	A14	RAS3	RAS2	RAS1	RAS0
1	1	0	0	1	1	1	0
1	1	0	1	1	1	0	1
1	1	1	0	1	0	1	1
1	1	1	1	0	1	1	1
1	0	X	X	0	0	0	0
0	X	X	X	1	1	1	1

Sygnał $\overline{CAS}$=0 (blok 3) generowany jest tylko w cyklu zapisu/odczytu pamięci przez procesor, jednocześnie dla całej matrycy. Opadające zbocze sygnału $\overline{CAS}$ powoduje:
— w cyklu zapisu — wpisanie informacji z szyny danych systemu do pamięci dynamicznej (jednocześnie sygnał $\overline{WRITE}$=0)
— w cyklu odczytu — podanie na szynę systemu danych z matrycy pamięci dynamicznej; aby dane zostały poprawnie zapisane do bufora musi to nastąpić przed upływem 180 ns od opadającego zbocza sygnału CAS.

Sygnał MUX (ang. Multiplexer) służy do przełączania adresu podawanego przez procesor, w sposób następujący:

MUX	Wyjścia Q multiplekserów U12, U13						
	Q6	Q5	Q4	Q3	Q2	Q1	Q0
0	A6	A5	A4	A3	A2	A1	A0
1	A13	A12	A12	A10	A9	A8	A7

Natomiast w cyklu odświeżania — sygnał $\overline{MEMACK}$=1, co powoduje że na matrycę pamięci podawany jest — przez multipleksery U14 i U15 — adres z liczników U16 i U17. Liczniki U16, U17 generują adres wierszy matrycy. Liczba wierszy wynosi 128. Nowy stan liczników ustalany jest przez opadające zbocze sygnału $\overline{REF}$. Wyjście adresowe multiplekserów U14, U15 doprowadzone są do matrycy pamięci przez oporniki 33Ω w celu zmniejszenia przesłuchów.



Ryc. 3. Schemat ideowy
(7400 — U1, U6; 7402 — U5; 7404 — U18, U19; 1/2 7437 — U7; 7474 — U2, U9; 7493 — U16, U17; 74138 — U11; 74157 — U12, U13, U14, U15; 74158 — U10; 74164 — U8; 74193 — U3, U4; 74374 — U20)

Schemat ideowy układu dekodera matrycy pamięci, multipleksera adresów, generatora adresu odświeżania przedstawiono na rysunku 3.

Bufor pamięci służy do pamiętania danych z matrycy pamięci podczas cyklu odczytu. Wpis do bufora dokonywany jest, gdy EOLN=1 i MEMACK=0. Natomiast sygnał EOLN = 0 powoduje:
— skasowanie żądania aktualnie wykonywanego cyklu w rejestrze zgłoszenia dostępu
— wyzerowanie rejestru przesuwnego 74174 (U8) —
— wycofanie wszystkich sygnałów strobujących do pamięci dynamicznej
— ustawienie sygnału READY=1, gdy wykonywany był cykl dostępu do pamięci przez procesor.

Zastosowanie bufora pośredniczącego pomiędzy szyną danych systemu a szyną DOUT pamięci pozwala na utrzymywanie danych podczas odczytu przez dowolnie długi okres. Dzięki temu pamięć dynamiczna widziana jest przez procesor jak zwykła pamięć statyczna.

Schemat ideowy bufora pamięci przedstawia rysunek 3 (blok 7). Zaproponowane rozwiązanie można uprościć stosując zamiast układów 1, 2, 3 automat synchroniczny z pamięcią PROM.

Uwaga: Nie należy stosować układu 8238 jako sterownika systemowego, ponieważ generuje on wyprzedzony sygnał zapisu do pamięci, co może prowadzić do błędów podczas zapisu informacji do pamięci przez procesor.

PIOTR SMÓLSKI
Przemysłowy Instytut Elektroniki
Warszawa

Programowanie w języku maszynowym jest żmudne. Trzeba nie lada wytrwałości i samozaparcia, aby napisać dłuższy program. Asembler rzadko jest wykorzystywany nawet przez profesjonalistów.

Do programowania mikrokomputerów najczęściej wykorzystywany jest BASIC. Jest to język tak prosty, iże osoba znająca angielski może opanować podstawowe zasady w ciągu zaledwie kilku godzin. Jednak po krótkim okresie fascynacji, użytkownika mikrokomputera zaczyna irytować długi czas oczekiwania na wynik realizacji programu. Coraz częściej okazuje się też, że nowo wymyślony program w żaden sposób nie daje się „upchnąć" w pamięci mikrokomputera...

Dlatego właśnie postanowiliśmy lansować wśród „członków" mikroKLANU język względnie wysokiego poziomu o nazwie FORTH. Za jego pomocą można uzyskać wyjątkowo zwięzły kod maszynowy (często porównywalny z programami pisanymi bezpośrednio w języku maszynowym!). Implementacje FORTH'A istnieją dla niemal wszystkich typów popularnych mikroprocesorów. Tak więc, niezależnie od posiadanego sprzętu, każdy może korzystać z tego języka!

Niebawem INFORMATYKA przedstawi obszerny opis FORTH'A, z zachowaniem pietyzmu należnego opisowi języka. W mikroKLANIE planujemy natomiast cykl adresowany do hobbystów. Po krótkim wprowadzeniu, mającym pomóc w myśleniu kategoriami FORTH'A, udostępnimy nasze łamy entuzjastom tego języka. FORTH da się lubić. I zapewne warto go polubić!

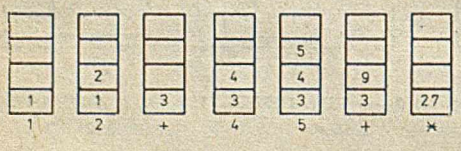
FORTH (1)

Stosy

Działania języka FORTH opierają się na przekazywaniu wszelkich parametrów przez stos. Język operuje głównie na stosie danych (LIFO, ang. Last In First Out) oraz na stosie adresowym podprogramów (RETURN-STACK). Można również odwoływać się do zmiennych prostych, stałych i tablic — lecz jest to wykonanie pewnych procedur, które swe działanie opierają na stosie.

Notacja odwrotna

FORTH używa standardowo odwrotnej notacji Łukasiewicza — znanej jako notacja polska. Opiera się ona na zasadzie, że najpierw podaje się argumenty, a następnie polecenie operacji (np. 1+2+4 = ? zapisujemy jako 1 2 4 + +). Przy wykonywaniu obliczenia (1+2)*(4+5) stan stosu przedstawia rysunek:



Słownik

Procedury języka FORTH nazywa się słowami i zapisuje się w słowniku. Podstawowy słownik standardu FORTH-79 zawiera 128 słów. Programowanie w tym języku polega na tworzeniu nowych słów, które zawierają pewną sekwencję odwołań do słów już istniejących. Przy wykonaniu danego słowa jego składowe są wywoływane kolejno jak procedury, a przekazywanie parametrów odbywa się za pośrednictwem stosu. Każde nowe słowo może być dołączone do słownika i wykorzystywane przy definiowaniu następnych. Można zatem tworzyć różne problemowe odmiany języka (np. do sterowania procesem, do grafiki, system operacyjny itd.).

Semikompilacja

W wyniku kompilacji nowego słowa pod nagłówkiem z nazwą tworzony jest spis adresów kolejnych słów (procedur, które składają się na dane słowo) oraz stałych, które mają być umieszczone na stosie. Wykonanie należy do interpretera, który przeszukuje słownik (porównując nazwy w nagłówkach) i przekazuje sterowanie do części wykonawczej danego słowa.

Przykładowy program

Możemy zdefiniować słowo BIN w sposób następujący:

```
:     (początek definicji)
BIN   (nazwa słowa)
2     (stała — na wierzchołek stosu)
BASE  (adres zmiennej BASE — na wierzchołek stosu. Zmienna BASE określa podstawę systemu liczbowego, w którym uzyskuje się wyniki)
!     (zapisanie pod adresem pobranym ze stosu (tj. adresem BASE) następnej wartości pobranej ze stosu (tj. stałej 2))
;     (koniec definicji)
```

Definiując analogicznie:

```
: OKT 8 BASE ! ;
: DEC 10 BASE ! ;
: HEX 16 BASE ! ;
```

możemy wykonać następujące obliczenia:

```
HEX   (ustalenie systemu szesnastkowego)
A     (wpisanie na stos stałej dziesięć)
DE    (powrót do systemu dziesiętnego)
.     (polecenie wydruku lub wyświetlenia ostatniej pozycji ze stosu)
CR    (koniec polecenia)
10    (wynik)
OK    (komunikat systemu o wykonaniu polecenia)
```

W podobny sposób wykonuje się poniższy ciąg rozkazów:

```
BIN 1111 DEC . CR 15 OK
BIN 1111 OKT . CR 17 OK
DEC 10 BIN . CR 1010 OK
HEX FF OKT . CR 377 OK
```

MAREK CZARZYŃSKI
ABAKUS WARSZAWA

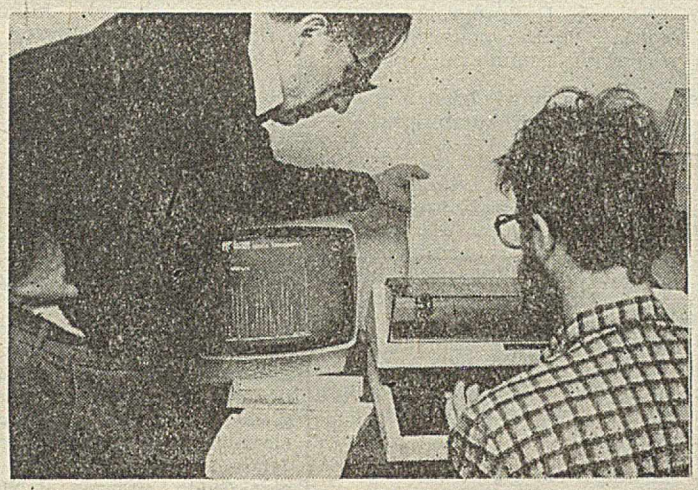
Nie licz na poparcie

Licz na mikrokomputerze!

Mikrokomputer BBC

Wywiad z posiadaczami BBC

— Andrzejem Pląskowskim, doktorem chemii, i Witoldem Pląskowskim, magistrem matematyki, którzy zaczynają prowadzić w Warszawie usługi komputerowe przy użyciu własnego mikrokomputera



µK: — BBC znaczy British Broadcasting Corporation i kojarzy się z angielską rozgłośnią radiową. Co to ma wspólnego z mikrokomputerem?

AWP: — Wyjaśnienie jest proste. Rozgłośnia BBC ogłosiła przetarg na produkcję seryjną mikrokomputera do celów dydaktycznych — z zamiarem masowego wykorzystania go we własnych programach radiowo-telewizyjnych. Przetarg wygrała firma ICORN ATOM i narodził się mikrokomputer BBC.

µK: — Cel rozgłośni jest wprawdzie ambitny, ale ma podłoże wyraźnie komercjalne. Zrozumiałe byłyby więc obawy przed takim zakupem. Gdy nie chce się człowiek poddawać panowaniu masowych środków przekazu...

AWP: — Przeciwnie. Od czasu zakupu mikrokomputera prawie nie oglądamy telewizji. Natomiast o zakupie komputera właśnie tej firmy zadecydowały jego walory użytkowe. Przed podjęciem ostatecznej decyzji prowadziliśmy długotrwałe konsultacje. Konkurencja na brytyjskim rynku mikrokomputerowym jest bardzo silna. W 1982 roku, gdy robiliśmy rozeznanie, bardzo popularny był mikrokomputer SHARP. Jednak znajomi radzili wstrzymać się z zakupem — aż do pojawienia się BBC.

µK: — Jakie są te walory użytkowe? Pozornie kształcącym, a faktycznie ogłupiającym grom telewizyjnym też przypisuje się szereg zalet...

AWP: — Nie w tym rzecz. Prawie wszystkie gry skasowaliśmy już po kilku tygodniach użytkowania, po prostu do gier go szkoda. BBC ma bardzo dobrze opracowany BASIC i najlepszą — zdaniem niektórych — grafikę na świecie (oczywiście w tej klasie sprzętu). Choć jest stosunkowo drogi, sam komputer kosztował 399 funtów.

µK: — To może kilka słów o samym komputerze.

AWP: — BBC jest oparty na mikroprocesorze typu 6502, ma 32 K słów pamięci użytkowej i 32 K słów przeznaczonych na pamięć stałą ROM. Możliwa jest współpraca z telewizorem systemu PAL lub SECAM, zależnie od zakupionego mikroobwodu — w ośmiu kolorach. Klawiatura jest typowa, QWERT, z dodatkowym rzędem klawiszy funkcyjnych, którym można na stałe przyporządkować dowolne funkcje (obejmujące jednak nie więcej niż 1 linię do 255 znaków). Oczywiście, w zestawie jest także drukarka mozaikowa (80 znaków w wierszu) i stacja dysków elastycznych, której niestety nie mamy. Jako pamięci masowej używamy magnetofonu kasetowego. Do wyposażenia rzadziej spotykanego w takim sprzęcie należą cztery wejścia analogowe i osiem dwustanowych wejść—wyjść cyfrowych. Translatory języków programowania są zawarte w pamięci ROM. Obecnie, oprócz BASICA, powinno być już w sprzedaży kilka innych „kostek", na przykład z PASCALEM.

µK: — Jakie są możliwości BASICA poza rozbudowaną grafiką?

AWP: — Oprócz typowych instrukcji, które ma każdy język, BBC BASIC ma instrukcje strukturalne:

REPEAT..UNTIL
FOR..NEXT
IF..THEN..ELSE
ON..GOTO
ON..GOSUB

i szereg funkcji standardowych, jak: sinus, cosinus, tangens i odpowiednie

- ZOINTE przy Instytucie Technologii Elektronowej wydał informację katalogową o wprowadzanym do produkcji przez CEMI zestawie układów mikroprocesorowych (MCY 7880, UCY 74S424 UCY 74S428). Nie byłoby nic nadzwyczajnego w fakcie, że producent informuje o parametrach własnego wyrobu, gdyby nie dotychczasowa praktyka CEMI. Informacja jest stosunkowo obszerna (zawiera nawet opis funkcjonalny — aż trudno uwierzyć!). Do ciemniejszych stron wydarzenia należy zaliczyć nakład (500, słownie: pięćset egzemplarzy!) i cenę zeszytów (np. 230 zł za opis CPU). Zastanawiający jest również fakt, że tym razem pominięto milczeniem (przynajmniej we wspomnianym wydawnictwie), jaki układ był pierwowzorem MCY 7880. Zdaje się, że konsekwencją „zabawy w chowanego" jest istne kuriozum: przetłumaczono na język polski mnemoniki wszystkich rozkazów. Pozostaje zapytać — dlaczego, dla lepszego kamuflażu, nie opracowano wersji ASSEMBLERA pismem klinowym? Byłaby niewątpliwie równie dogodna dla nieźle rozpowszechnionego już w kraju oprogramowania. I jeszcze próbka tłumaczenia: INTEL-owski mnemonik CNZ (ang. Call if Not Zero) to w „polskiej" wersji — KNZ (chyba od Koll jeśli Nie Zero). No cóż, nie będziemy kollać: autor! autor!

- Klub Mikrokomputerowy ABAKUS ogłosił wyniki konkursu na pomysł gry edukacyjnej. Przyznano trzy nagrody (mikrokomputer ZX81-1K, telewizor VELA, magnetofon kasetowy) i osiem wyróżnień (książki o ZX81 w języku angielskim).

  Oto laureaci:

  1. Dr Barbara Chrzan-Feluch (Instytut Matematyki Uniwersytetu Warszawskiego) — za zbiór pomysłów pt. „Matematyka w grach" dla dzieci w wieku 7—16 lat. Pomysły dotyczyły: tabliczki mnożenia, obliczania wielkości kątów i pól figur, orientacji w układzie współrzędnych.
  2. Andrzej Więckowski — za pomysł pt. „Biuro Matrymonialne". Gra jest zbliżona koncepcyjnie do popularnych niegdyś „Dyrektorów". Jej zadaniem jest symulacja warunków rozwoju małej firmy. (A. Więckowski otrzymał także wyróżnienie za pomysł „Sieć bramek logicznych").
  3. Pracownia Zastosowań Naukowo Technicznych ZETO—Wrocław — za pomysł pt. „Kolorowe nutki". Gra ma służyć do elementarnej edukacji muzycznej: określania wysokości dźwięku w gamie.

- Do 31 maja br. można nadsyłać prace w ogłoszonym przez klub ABAKUS konkursie na program gry edukacyjnej. Pierwszą nagrodę stanowi mikrokomputer ZX SPECTRUM! Ogłoszony został również konkurs wewnątrz-klubowy (ze względu na charakter nagrody) na program dla dzieci... w wieku 1—2 lat. Laureat otrzyma 1000 punktów, co w rozliczeniach klubowych równoważne jest np. możliwości korzystania z ZX SPECTRUM przez 500 godzin.

- 28 grudnia ub.r. klub ABAKUS liczył 110 członków. Klub czynny jest codziennie w godzinach 14—20.

- W krakowskiej „Piwnicy pod Baranami" również powstał klub mikrokomputerowy.

- W połowie lutego w warszawskim Pałacu Młodzieży (PKiN) odbędzie się Komputerowy Turniej Szachowy. Uczestnicy będą mogli grać przeciwko komputerowi, lub — za pośrednictwem komputera — w parach przezeń skojarzonych. Zapisy partii będą oczywiście drukowane przez komputer. Współorganizatorem Turnieju jest klub ABAKUS.

- W dniach 12—27 listopada ub.r. w Muzeum Techniki odbyła się wystawa pt. „Komputer dla każdego". Prezentowano wyroby firm parających się techniką mikroprocesorową. Rewelacji nie było. Prym wiódł ABAKUS.

- Powstał zapowiadany (Informatyka nr 5/83) Klub Użytkowników Mikroprocesorów (KUM) — zgłosiło się ponad 100 osób. Przypominamy: KUM zrzesza profesjonalistów, amatorów i instytucje. Celem działania klubu jest wymiana informacji i usług. Planowane jest założenie biblioteki oprogramowania w myśl zasady: wkładasz do wspólnej puli swoją umowną jednostkę programową — otrzymujesz inny, potrzebny ci program. Przewidywane są specjalne premie za wykrycie ewentualnych błędów i usprawnienia programów bibliotecznych.

  Klub zamierza wydawać coroczne spisy nazw, nazwisk, adresów, telefonów wraz z omówieniem sposobu korzystania z techniki mikroprocesorowej — dla wszystkich członków. Będą także periodycznie rozsyłane oferty produktów i usług (na koszt ogłaszającego) do wszystkich zrzeszonych. W biuletynie będzie można również zamieścić prośbę o pomoc w rozwiązaniu określonego problemu. KUM rozpoczął rozmowy w sprawie preferencyjnego traktowania potrzeb członków klubu.

  Miejmy nadzieję, że nie będzie się to odbywać kosztem innych amatorów, nie zrzeszonych w klubie.
  Dla minimalizacji obsługi administracyjnej założono komputerową bazę danych ułatwiającą odnalezienie żądanej informacji, jak również przygotowywanie spisów i biuletynów.
  Do klubu można przystąpić wypełniając ankietę i przesyłając niewielkie wpisowe. Ankietę klubową można otrzymać od założyciela klubu — Andrzeja Drożniaka; ul. Pytlasińskiego 14 m. 4, 00-707 WARSZAWA; tel. dom. 41-26-01.

- **Informacje mikroKLANU prosimy przesyłać bez zwłoki — pod adresem INFORMATYKI: 00-041 Warszawa, ul. Jasna 14/16, p. 244.**

funkcje odwrotne (arecin, arccos, arctan), pierwiastek kwadratowy, logarytm naturalny i dziesiętny, funkcja wykładnicza, moduł, funkcje logiczne AND, OR i NOT, obliczenia stopni kątowch, funkcja losowa i odczyt wejścia analogowego. Liczby w BASICU mogą mieć 10 cyfr znaczących i znak. Dużą zaletą translatora jest jego szybkość. Do programów w BASICU można włączać podprogramy napisane w języku asemblera.

µK. — A co można powiedzieć o wadach czy ograniczeniach komputera?

AWP: — Jego słabą stroną jest drukarka i klawiatura. Poza tym system załamuje się przy dużych zbiorach. W praktyce długość programu nie może przekroczyć 5000 linii przy operowaniu liczbami całkowitymi i 3000 przy liczbach rzeczywistych. Tego ograniczenia unika się, oczywiście, przy pracy z dyskiem elastycznym. Ze względu na dość mały wymiar strony tekstu, stosunkowo niewygodnie pracuje się z telewizorem o dużym ekranie. Obecnie używamy Neptuna 150.

µK: — Zgłębiliście Panowie już wszystkie możliwości BBC?

AWP: — Skądże, do tego nam daleko. Intensywnie pracujemy dopiero od kilku miesięcy.

µK: — A jaki cel Panom w tym przyświeca?

AP: — Obaj pracowaliśmy w instytutach naukowych. Po sprowadzeniu mikrokomputera, rozwiązałem przy jego użyciu pewien niewielki problem. Syn, z zawodu matematyk, stwierdził: „Tata, przecież to można zrobić dużo prościej". I tak się zaczęło. Obecnie używamy go do różnych obliczeń w zagadnieniach praktycznych, np. w określaniu właściwości wybuchów, w pomiarach natężenia ścieków metodą korelacyjną, w częstotliwościowych metodach identyfikacji...

µK: — Z jakim skutkiem?

AP: — Zadziwiająco pozytywnym. W automatyzacji wielu nawet skomplikowanych obliczeń ten mikrokomputer jest niezastąpiony, znakomicie wyręcza człowieka. Co ciekawe, ta technika promieniuje. Córka, z zawodu zootechnik, zaczęła go używać do obliczeń statystycznych, kilka instytucji po nawiązaniu współpracy z nami nosi się z zamiarem zakupienia BBC. W innych przypadkach, na podstawie naszych opracowań lub założeń, realizuje się zamówione w firmach krajowych wyspecjalizowane urządzenia oparte na mikroprocesorach.

Spośród szeregu podręczników dotyczących mikrokomputera BBC warto wymienić następujce:

Angell O., Jones B. J.: Advanced Graphics with the BBC Microcomputer. Macmillan Press, London, 1982
Birnbaum J.: Assembly Language Programming for the BBC Microcomputer. Macmillan Press, London, 1982
James M.: The BBC Micro — An Expert Guide. Granada Publishing, London, 1983
Scanlon L. J.: 6502 Software Design. Howard Sams, Indianapolis (IN), 1980.

µK. — Chyba wynika to ze specyficznej sytuacji naszego rynku komputerowego, tzn. braków i cen.

AP: — Nie tylko. Instytut w Manchester gdzie przebywałem zakupił kilkanaście sztuk do celów wyłącznie naukowych.

µK: — Dziękuję za rozmowę. Życzymy Panom jeszcze większych postępów z BBC. A łamy mikroKLANU czekają na opis ciekawszych rozwiązań.

Rozmowę prowadził
JANUSZ ZALEWSKI

Absolwenci, amatorzy, cudotwórcy, entuzjaści, fachowcy, konstruktorzy, naukowcy, neofici, nonkonformiści, odkrywcy, pasjonaci, potentaci, profesjonaliści, programiści, prowodyrzy, rekordziści, sceptycy, studenci, uczniowie, użytkownicy, wyrobnicy — spróbujcie z nami...

INFORMATYKA tel. 27-71-40

ABAKUS tel. 42-91-85

KUM tel. 41-26-01

µK

prowadzi
Andrzej J. Piotrowski
tel. dom. 48-22-85